**Softwareontwerp & -architectuur 3 Opdracht 1**

# Inleiding

Opdracht ter voorbereiding op college 2.

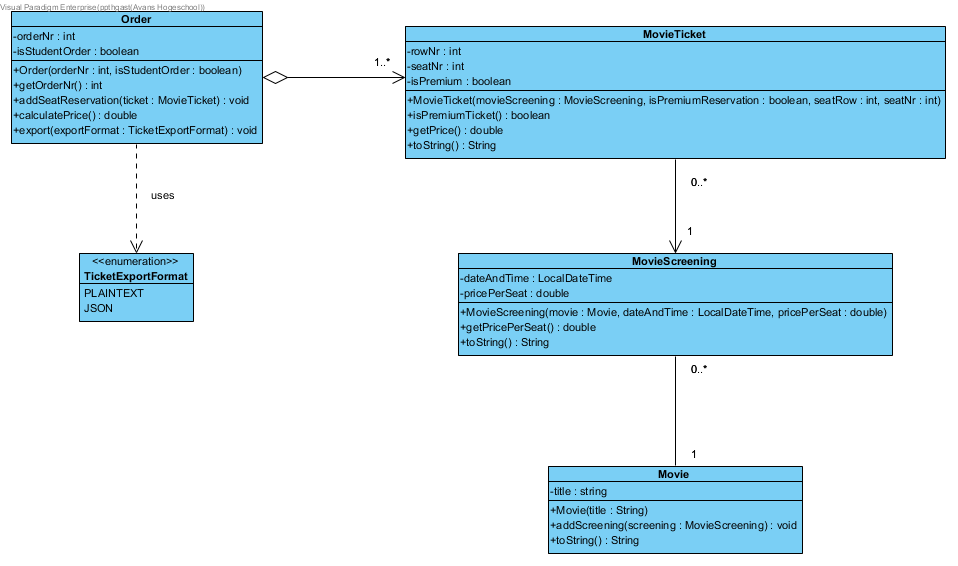
Voor een bioscoopsysteem gaan we een paar klassen ontwikkelen. In de bioscoop draaien natuurlijk films. Elke film wordt minimaal 1, maar in veel gevallen vaker vertoond. Een bezoeker kan kaartjes kopen voor een voorstelling van een film en daarbij kan deze aangeven of hij/zij een premium zitplaats wil. De stoel is dan wat luxer uitgevoerd, zit natuurlijk comfortabeler en heeft net wat handiger vakjes voor eten en drinken dan de standaardstoel. Een bestelling kan bestaan uit zowel gewone- als premium-stoelen. Dat zal niet zijn wanneer je met twee personen gaat, maar als je met een groep gaat en je wilt bij elkaar in de buurt zitten, kan zo’n mix ontstaan.

Wat de prijsberekening betreft, is er een aantal regels (die voor studenten voordelig uitpakken):

* Elk 2e ticket is gratis voor studenten (elke dag van de week) of als het een voorstelling betreft op een doordeweekse dag (ma/di/wo/do) voor iedereen.
* In het weekend betaal je als niet-student de volle prijs, tenzij de bestelling uit 6 kaartjes of meer bestaat, dan krijg je 10% groepskorting.
* Een premium ticket is voor studenten 2,- duurder dan de standaardprijs per stoel van de voorstelling, voor niet-studenten 3,-. Deze worden in de kortingen verrekend (dus bij een 2e ticket dat gratis is, ook geen extra kosten; bij 10% korting ook 10% van de extra kosten).
* Om de casus niet nog complexer te maken, gaan we ervan uit dat bij een studenten-order alle tickets voor studenten zijn; vandaar het isStudentOrder attribuut in de Order klasse en niet in MovieTicket.  
  (is wel een leuke uitdaging om deze eis er wél in op te nemen).

Voor de toekomst voorziet de bioscoop nog andere doelgroepen met andere kortingen.

Een Order kun je ook exporteren. Voorlopig in 2 formaten naar file, plain text en JSON formaat.



# Opdracht

* Je programmeert het algoritme voor de prijsberekening in de calculatePrice methode van Order.
* Je programmeert de export-functionaliteit in de export methode van Order.
* Je bedenkt test cases voor de prijsberekening en programmeert deze. Laat zien hoe je aan de test cases komt.
* Je mag programmeren in een taal naar keuze; de docent werkt zijn/haar voorbeelden uit in Java.
* Je neemt jouw uitwerking op in de Azure Devops omgeving, en maakt daarnaast ook een development pipeline die automatisch de unit testen runt. Doe dat op de master-branch, omdat we later tooling gaan gebruiken die in de gratis versie alleen op de master-branch werkt.

Je werkt in duo’s; dezelfde duo’s als voor de eindopdracht. Zorg voor een Azure DevOps omgeving of equivalent en daar voeg je je uitwerking aan toe in een map ‘Opdracht 1’. Voeg een tekst- of Word-document toe waarin je kort beschrijft hoe je jouw set van testcases hebt bepaald. In college 2 gaan we een aantal uitwerkingen van de algoritmes vergelijken en gaan we verder in op het definiëren van de test cases.